



Consultando con filtros

Transcripción

[00:00] Hola. Continuando con las consultas, en esta parte nosotros vamos a realizar las consultas con filtro. Vamos a realizar las consultas de todos los elementos en la tabla producto que tenga un determinado nombre y vamos a realizar la consulta de todos los elementos en la tabla producto, que tenga un determinado nombre de una categoría.

[00:18] Recordamos que nosotros tenemos una relación entre la tabla producto y la tabla categoría, y nosotros vamos a buscar esta vez los elementos en la tabla producto que comiencen por el nombre de la categoría. Entonces vamos a ir a nuestra clase Dao, donde vamos a comenzar agregar esos métodos.

[00:37] El primer método que va a ser un método público, nos va a retornar una lista en la entidad producto. Se va a llamar consultaPorNombre. Tiene que recibir un parámetro. Ese parámetro va a ser del tipo string y va a ser el nombre.

[00:56] Ahora, yo tengo que establecer mi consulta. Esa consulta, como ya lo habíamos mencionado, va a ser un string, que nosotros vamos a establecer, que la vamos a llamar jpql, y vamos a hacer "SELECT", el parámetro P FROM la entidad Producto con el token P.

[01:25] Y a partir de este punto nosotros comenzamos a realizar el filtro. Con la palabra WHERE, que es usada en SQL, en este caso sería JPQL, WHERE P.nombre=nombre que va a ser el parámetro que vamos a establecer. Y aquí

realizamos el retorno, em. creamos la consulta, createQuery, vamos a pasar la consulta y tenemos que establecer el parámetro que vamos a buscar.

[02:03] Entonces, vamos a utilizar el método setParameter y nos indica que tenemos que indicar la posición y el valor. La posición en este caso sería nombre y el valor sería nombre, que es el que estoy pasando como parámetro. Luego de esto voy a tener una lista de resultados, al igual que nosotros habíamos hecho de la consulta de todos los elementos.

[02:28] Entonces solamente aquí voy a mencionar, que si yo quiero agregar múltiples parámetros, entonces yo puedo utilizar la palabra AND y continuar agregando más parámetros. Por ejemplo, P descripción y aquí colocaríamos igual descripción. Entonces podemos establecer todos los parámetros que nosotros queramos.

[02:57] Y nosotros podríamos sustituir el nombre por el valor numérico, colocaríamos el valor 1, aquí el valor 2, y en setParameter reemplazaríamos ese valor del nombre por la posición numérica. Nosotros lo vamos a dejar con la palabra y vamos a establecer nuestra consulta.

[03:26] Vamos a ir a RegistroDeProducto, no vamos a realizar una consulta de todos los elementos. Ahora vamos a realizar una consulta por nombre. Entonces sería consulta por nombre. ¿Cuál sería el nombre que nosotros queremos buscar? Sería este de acá.

[03:43] Entonces vamos a copiar de acá el nombre del producto, lo pegamos acá y deberíamos obtener la descripción. Sería acá "Muy legal". Vamos a imprimir el resultado en la consola. Guardamos. Entonces, vemos que efectivamente tenemos la descripción para el elemento producto.

[04:12] Como teníamos únicamente un único valor en la tabla solamente nos mostró un resultado. Para el siguiente valor, el siguiente método que nosotros vamos a hablar es consulta por nombre de categoría. Entonces ese también va

a ser un método público que nos va hablar como retorno una lista de la entidad producto, se va a llamar `consultaPorNombreDeCategoría`.

[04:44] Va a recibir como parámetro el nombre de la categoría. Entonces, también va a ser un elemento del tipo de string, vamos a llamar nombre. Aquí vamos a colocar la consulta que sería `String jpql` y vamos a colocar, de la misma forma que hicimos acá, vamos a colocar sería `"SELECT p FROM Producto AS p WHERE"`; entonces en esta parte viene la diferencia.

[05:37] No sería el nombre sino la categoría. Y para buscar el nombre de esa categoría, simplemente colocar un punto e ir al nombre de esa categoría. Entonces él va a realizar esa secuencia. Va a entrar en la entidad p, va a continuar con la categoría de la entidad p y posteriormente va a ir al nombre.

[06:02] Entonces, vamos a colocar aquí igual al nombre que estamos pasando como parámetro. Entonces, de la misma forma que hicimos anteriormente, vamos a colocar acá un `return` en, realizamos la consulta y enviamos la consulta acá como parámetro.

[06:28] Vamos a establecer nuestro parámetro. Va a ser `"nombre"` y el valor para acá va a ser nombre. Vamos a tener una lista para ese método. Entonces guardamos acá, vamos a la clase `RegistroDeProducto` y vamos a reemplazar acá nuestro método `consultarPorNombre` por `consultarPorNombreDeCategoría` y tenemos que pasarle el nombre de la categoría que sería el nombre `"CELULARES"`.

[07:03] Lo ponemos acá, vamos a alterar acá, vamos a colocar `"Muy bueno"` y vamos a ejecutar esta aplicación. Vemos que nos dio como resultado el valor que nosotros colocamos en la descripción y con eso tenemos los dos métodos de filtrado, que sería métodos de consulta por nombre y método de consulta por nombre de categoría.

[07:32] La siguiente parte nosotros no vamos a obtener todos los elementos sino vamos a limitar el resultado únicamente a una de las propiedades o

columnas de esa entidad y vamos a agregar una nueva propiedad dentro de nuestro archivo de persistencia que nos va a permitir dar un mejor formato, una mejor visualización para este resultado que estamos obteniendo aquí.

[07:59] Y si nosotros analizamos realmente esas consultas, vemos que él realizó un select, vemos que colocó producto, fecha de registro, él buscó todos los elementos y si vemos acá, él realizó un cross join con la entidad categoría y aquí en la parte superior él realizó un outer join.

[08:26] Entonces, vemos que él está efectivamente haciendo las relaciones de forma automática. Nosotros no colocamos eso dentro de nuestra consulta, sino que JPA se encarga de realizar todas esas consultas en SQL de forma automática. Nosotros solamente le tenemos que indicar mediante el lenguaje de JPQL cuáles son los valores que queremos obtener.

[08:47] Entonces, en el siguiente video nosotros vamos a ver las consultas limitadas a un único elemento y vamos a agregar esa nueva propiedad.